

## Check and Validate input field when loses focus, with PHP via Ajax

This tutorial shows how to **Check and Validate** input form field when loses focus (onblur event), with PHP via Ajax.

- First, we create the HTML form. Each input field must be added into a <label>, <span>, or <div> element, together with a <span> tag with class="err", in this item it is added the error message associated to that input field.

- Example with a form with 3 input text fields, for Name, Password and E-mail:

```
<form action="some_script.php" method="post">
  <label for="exname">Name: <input type="text" id="exname" name="exname"
/><span class="err"></span></label>
  <div>Password: <input type="password" id="expass" name="expass" /><span
class="err"></span></div>
  <label for="exmail">E-mail: <input type="text" id="exmail" name="exmail"
/><span class="err"></span></label><br/>
  <input type="submit" class="exsbm" disabled="disabled" value="Send" />
</form>
```

The idea is that both the input field and the item with class="err" must be added into a parent element, usually <label>, but can also be a <div> (like the password field from this example).

The "submit" button is initially disabled, it is enabled after the input fields are validated.

- Then we use the following JavaScript object with an Ajax function that is called when the input field loses focus (when the "onblur" event is triggered). The Ajax function sends the name and the value to a PHP script that checks and validates the received value.

If the response from the PHP script is not an empty string, means the field is not valid, and the JavaScript object changes the CSS class of the input field, and adds the PHP response in the item with: class="err" associated to that field.

If the response from PHP is an empty string, means the field is valid, and the JavaScript object sets other class to that input field, empties the tag with class="err", and enables the "submit" button if there is no other invalid input.

- In this example the Submit button has class="exsbm"; if you change this class in the Submit button, you must change it in the JavaScript code too, in the setelm() function.

- To register the "onblur" event to the input field you want to be validated, you can add: onblur="chkF.checkAjax(this);" in that <input> tag, or this instruction in the JavaScript code (see the comments in the script presented bellow):

```
document.getElementById('field_id').onblur = function() {
chkF.checkAjax(this); }
```

### JavaScript script with the Ajax function to check and validate form field

```
<script type="text/javascript">
// object to validate form fields when they lose focus, via Ajax
var checkFormElm = function() {
  // from: http://coursesweb.net/ajax/
  var phpcheck = 'check.php'; // Here add the php file that validate
the form element
  var err = {}; // stores form elements name, with value of 1 for
invalid, and value 0 for valid
```

```

// change the css class of elm
var setelm = function(elm, val) {
    // if val not empty, sets in err an item with element name, and value of
1
    // sets border to this form element,
    // else, sets in err an item with element name, and value of 0, and
removes the border
    if(val != '') {
        err[elm.name] = 1;
        elm.className = 'errinp';
        if(elm.form.querySelector('.exsbm'))
elm.form.querySelector('.exsbm').setAttribute('disabled', 'disabled');
// disables the submit
        elm.parentNode.querySelector('.err').innerHTML = val;    // adds the
error message
    }
    else {
        err[elm.name] = 0;
        elm.className = 'vinp';
        elm.parentNode.querySelector('.err').innerHTML = '';    // removes
the error message

        // check if invalid items in $err (with value 1)
        var inv = 0;
        for (var key in err) {
            if(err[key] == 1) {
                inv = 1;
                break;
            }
        }

        // if not invalid element, enables the submit button
        if(inv == 0 && elm.form.querySelector('.exsbm'))
elm.form.querySelector('.exsbm').removeAttribute('disabled');
    }
}

// sends data to a php file, via POST, and displays the received answer
this.checkAjax = function(elm) {
    if(elm.value != '') {
        var xmlhttp = (window.ActiveXObject) ? new
ActiveXObject("Microsoft.XMLHTTP") : new XMLHttpRequest();    // gets the
XMLHttpRequest instance

        // create pairs index=value with data that must be sent to server
        var datatosend = elm.name +'='+ elm.value;

        xmlhttp.open("POST", phpcheck, true);    // set the request to php
file

        // adds a header to tell the PHP script to recognize the data as is
sent via POST
        xmlhttp.setRequestHeader("Content-type", "application/x-www-form-
urlencoded");
        xmlhttp.send(datatosend);    // calls the send() method with datas as
parameter

        // Check request status
        xmlhttp.onreadystatechange = function() {
            if (xmlhttp.readyState == 4) setelm(elm, xmlhttp.responseText);
        }
    }
}

```

```

    }
    else setelm(elm, 'This field must be completed.');
```

```

}

var chkF = new checkFormElm(); // object instance of
checkFormElm()

// HERE register onblur event to form elements that must be validated with
PHP via Ajax
document.getElementById('exname').onblur = function() {
chkF.checkAjax(this); }
document.getElementById('expass').onblur = function() {
chkF.checkAjax(this); }
document.getElementById('exmail').onblur = function() {
chkF.checkAjax(this); }
</script>
```

- In this example, the PHP script that checks and validates the form field value is added in "check.php" file.

Here is a php code that validate the Name, Password, and E-mail. You can create the PHP script to check the input field value according to your needs. The PHP response must be an empty string when the value is valid.

```

<?php
// http://coursesweb.net/php-mysql/

// array with error messages for each input field
$error = array(
    'exname'=>'The name must contain minimum 3 characters, letters, numbers
and space',
    'expass'=>'The password must contain between 7 and 15 characters.',
    'exmail'=>'Invalid email address'
);
$return = ''; // data to return

// validate the POST value
if(isset($_POST['exname']) && !preg_match('/^[a-z0-9 ]{3,}$/i',
$_POST['exname'])) $return = $error['exname'];
else if(isset($_POST['expass']) && (strlen($_POST['expass']) < 7 ||
strlen($_POST['expass']) > 15)) $return = $error['expass'];
else if(isset($_POST['exmail']) && !preg_match('/^[a-z0-9-]+(\.[a-z0-9-
]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*(\.a-z){2,4}$/i', $_POST['exmail'])) $return
= $error['exmail'];

echo $return;
```

- Now we set CSS code to style the error message, and the input fields for when they are invalid, and validated:

```

/* style for validated field */
.vinp {
background: #efeffe;
border: 1px solid blue;
}
```

```
/* style for invalided field */
.errinp {
  background: #fefbda;
  border: 2px dashed red;
}

/* style for error message */
.err {
  margin: 0 5px;
  color: red;
  font-style: italic;
}
```